Overview

Although Github Action has many features, we will only be focusing on CI/CD. We will be using two files integrate.yml and deploy.yml for integrating and deploying, respectively. They are under .github/workflows if you want to take a look. Whenever you make a pull request on the master branch the integration process will start, and whenever you push to the master branch the project will be automatically deployed.

We will be using NodeJs and the package Mocha to run our tests on top of a ubuntu container using Github Action. We are using webpack to build our JavaScript project and firebase to deploy the project.

Important: Never push your code directly to the Master branch. We currently have a branch called Develop; push your code there and create a pull request and only merge the pull request to the master branch if all the tests pass. Optionally, you can create your own branch.

Note: If you want to incorporate another piece of technology to the workflow, please contact a lead or someone in the build team.

To Get Started

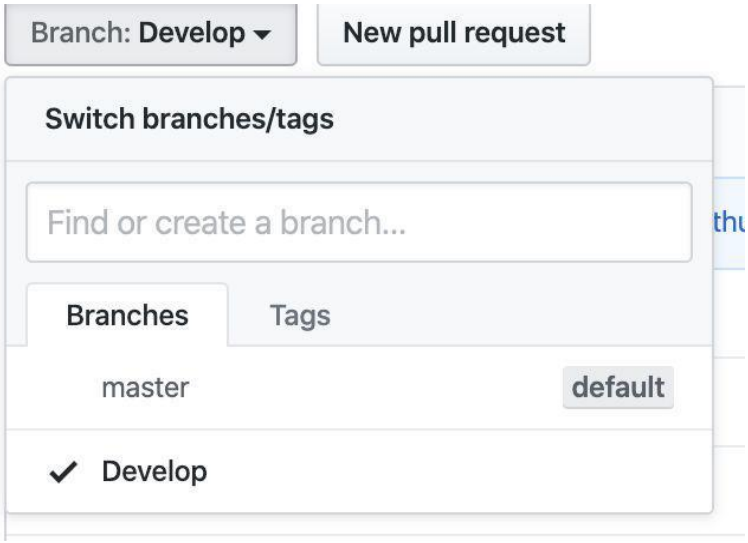
1. Make sure you have NodeJs installed, we are using version 12
    a. https://nodejs.org/en/download/

2. Run npm install in the root directory of the repository to install all the dependency in package.json. The dependency be installed in ./node_module in the repository. This should already be done for you, but if we ever add new dependency you need to run this again.

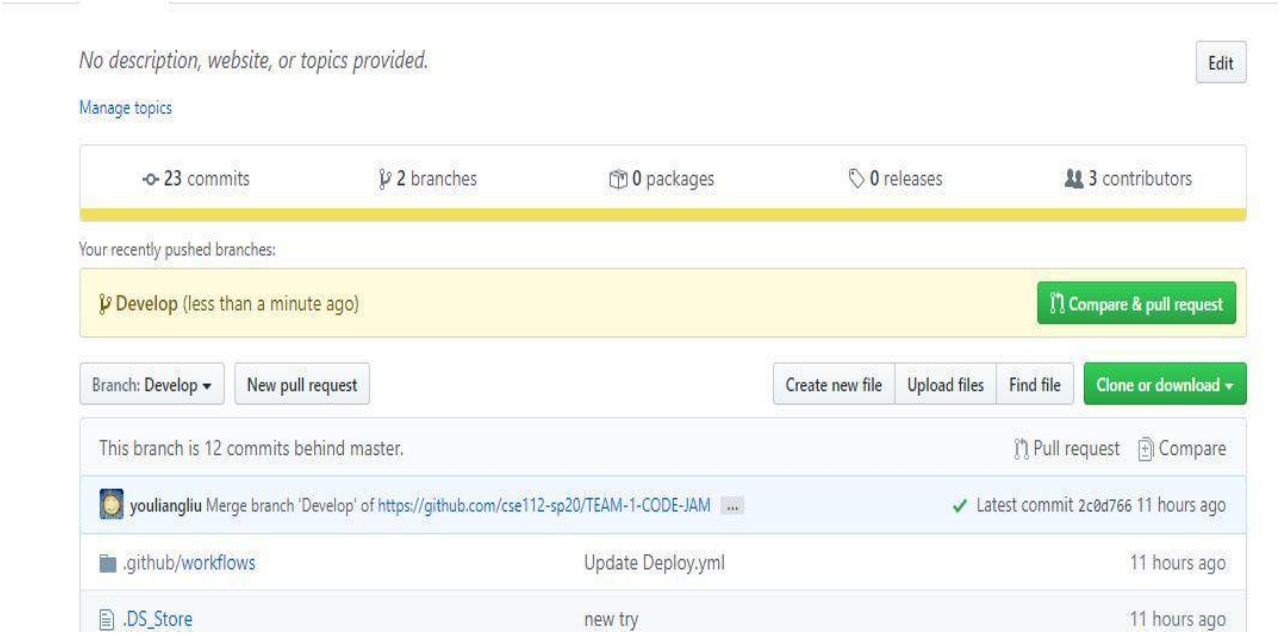
3. To test your code locally, run npm run test. For inspiration look at helloWorld.js and ./test/helloWorld.test.js. To Build your cold locally run npm run build. Note the pipeline does all of this for you already. So you can simply do a git push in a non-master branch and make a pull request to the master branch.

4. Run git checkout Develop in the command line, or checkout a non-master branch of your choice.

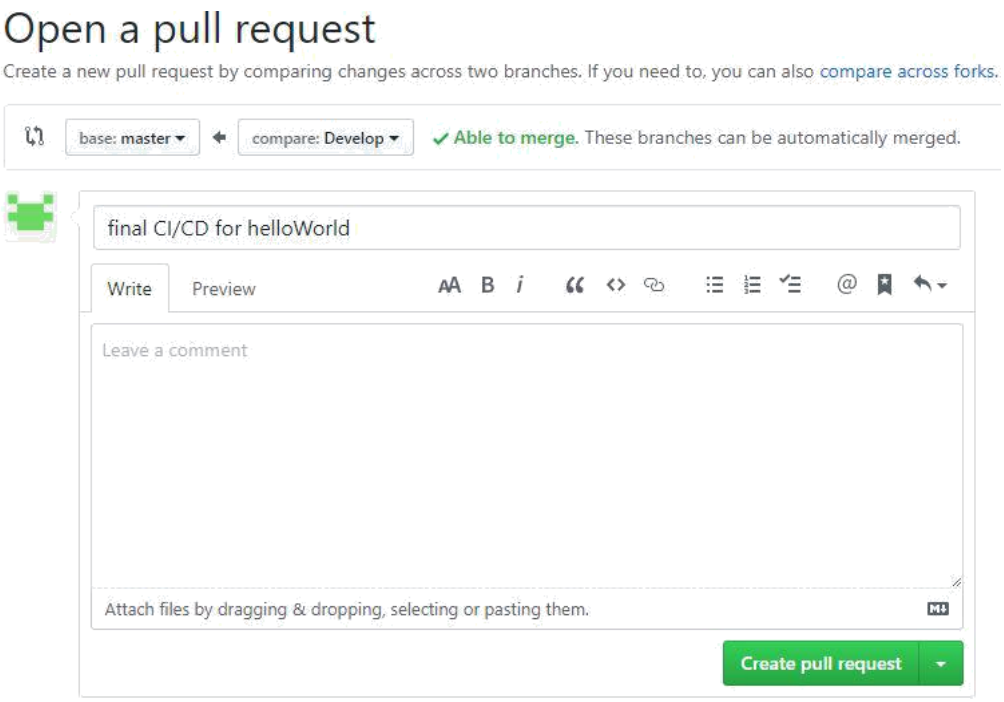
5. Push code to the Develop branch. (Or a non-master branch of your choice) You can switch the view of branches in github like so.



6. Next create a pull request to the master branch. If you push to a non-master branch Compare & pull requestbutton should appear. Click on it.



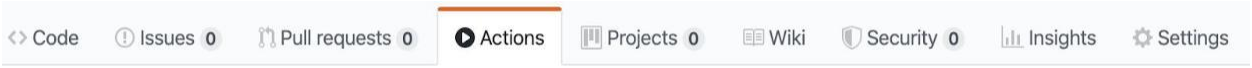
Alternatively, click on the New pull requestbutton.



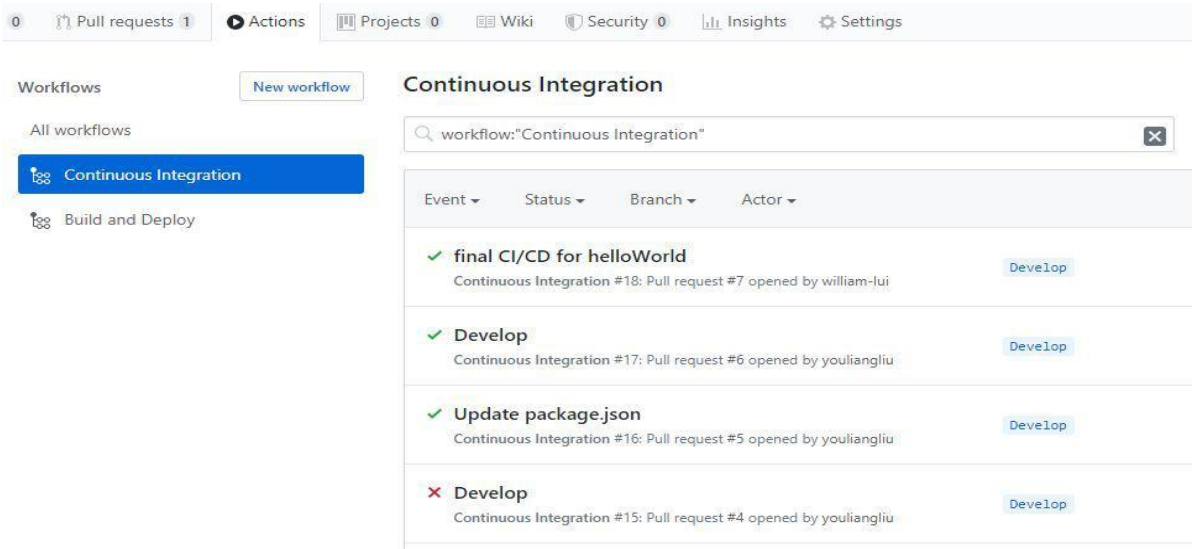
Make sure base is master, and compare is Develop

7. After creating a pull request, GitHub Action will automatically run all the test files in the push (Note: All test files should be in the test directory).

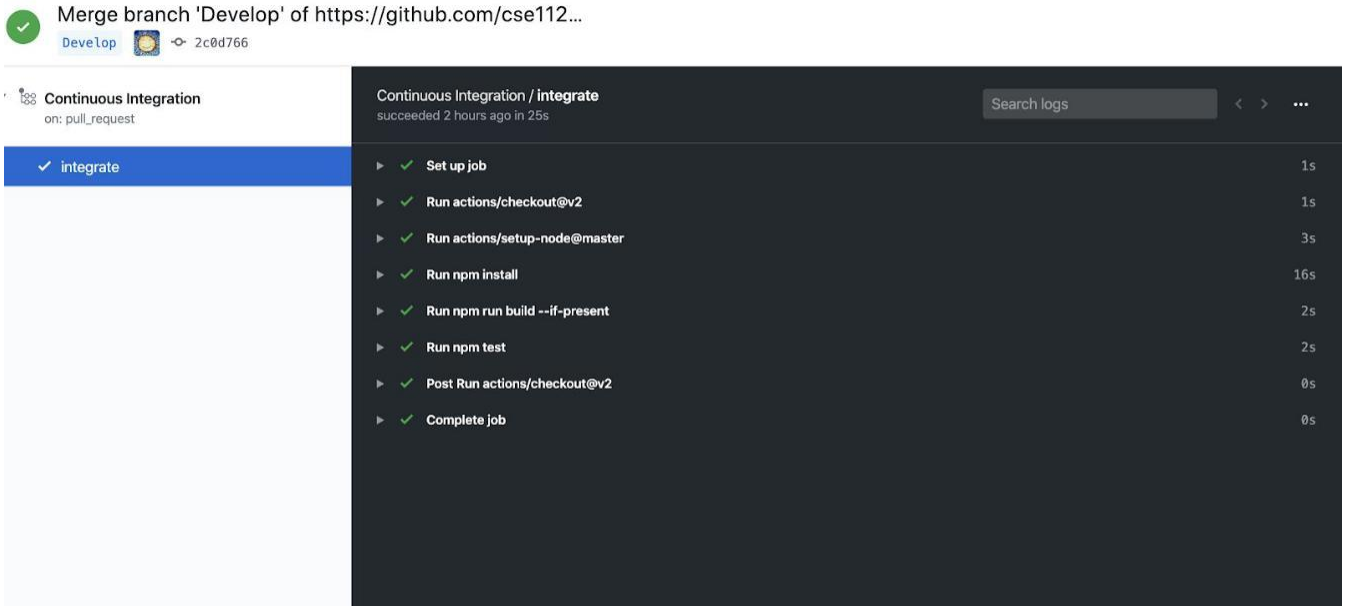
   Click on Actions in the toolbar.



   Click on the “Continuous Integration” category and click on the specific commit to view the test result.

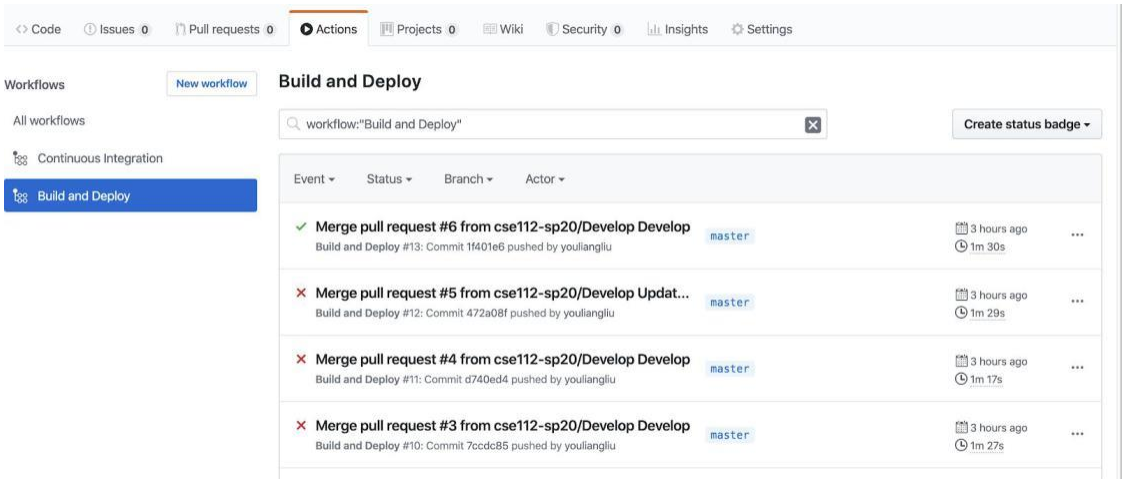


8. Tests that failed are labeled with ✗, and tests that passed are labeled with ✓. If we want to see the result from the commit “final CI/CD for helloWorld”. Click on it to see the result of each specific step of the integration process. In case any of the steps failed, you can click on the step to see the error message.

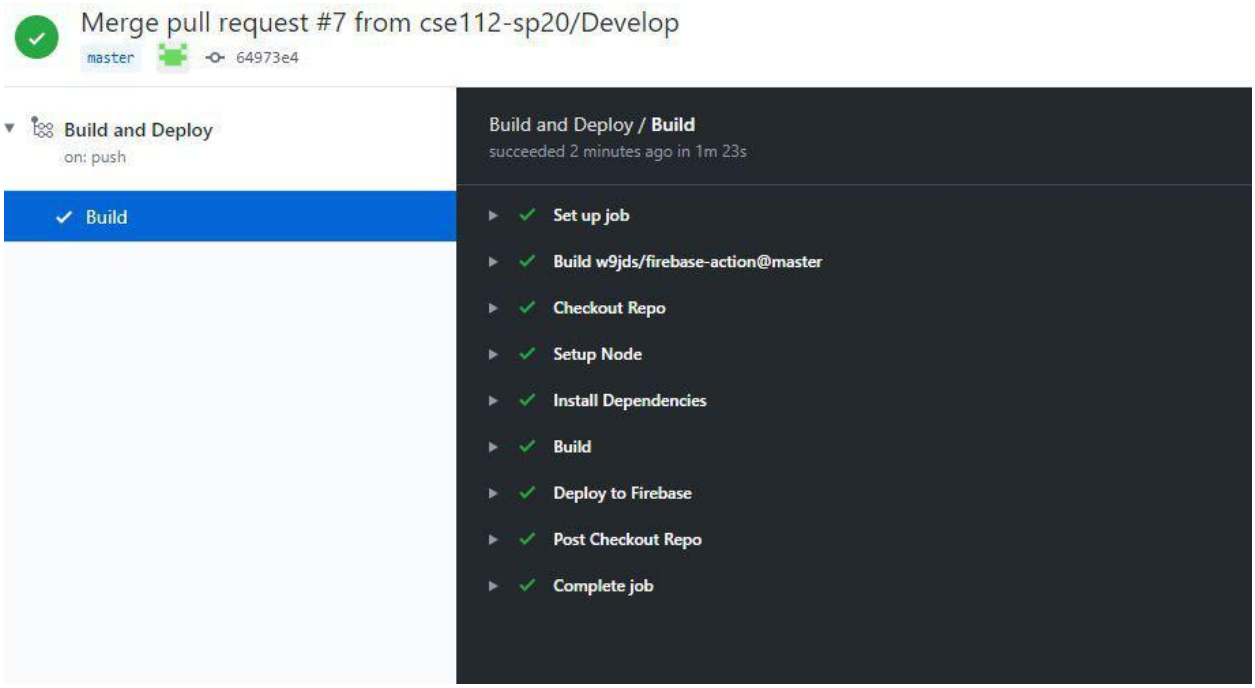


Note: if any of the tests fail and you want to push again to rerun the test, you don’t have to create a new pull request. The one from earlier with the failure will remain open. The branch will close when you either do it manually or merge the branch with master.

9. If all the tests passed, you can choose to merge to the master branch. Similar to the integration process. You can view the build and deploy process in the Actions toolbar under category “Build and Deploy”.



10. The code will then automatically be deployed to Firebase if all the test passes.



Features

Feature	Description
Release	
Auto Deploy	Deploy your application to a production environment immediately on push.
AWS support https://github.com/marketplace/actions/amplify-cli-action	Can use Aws Amplify to deploy our software.
Firebase support https://github.com/marketplace/actions/github-action-for-firebase	Can use Firebase to deploy our software
Auto Build	Before deploying, it will automatically build your code in its container
Docker support	Maintain Docker-based projects

Platforms	
Simultaneous build	Can build across different platforms at the same time
Cross-platform	Have build support for windows, mac, and linux
Framework version management	Can run a different version of the same framework, for example Node, across different platform
Test	
Dependency Scanning	Analyze your dependencies, and check if they are correctly defined.
Multiple Testing Framework	Supports many testing frameworks, for example jest and mocha.
Javascript	
Sass Build https://github.com/marketplace/actions/sass-build	JavaScript wrapper runs Sass build with provided Inputs
Generate JSDoc https://github.com/marketplace/actions/generate-jsdoc-documentation	build JSDoc docs from a defined source directory
AutoCheck JavaScript tests https://github.com/marketplace/actions/autocheck-javascript-tests-by-testomatio	shows changed tests on each pull request with a complete list of all tests in this project.
Check Deprecated Dependencies https://github.com/marketplace/actions/check-deprecated-dependencies	This action checks for all project dependencies and fails the action if a deprecated dependency found
Miscellaneous	
Slack Notification https://github.com/marketplace/actions/slack-notification	This action can be used to send messages about the status of a Git Action workflow.
Danger JS https://github.com/marketplace/actions/danger-js-action	Danger can help lint your rote tasks in daily code review.You can use Danger to codify your teams norms.

FAQs

How can I deploy my current code?

Push your code into the repository and make a pull request. Once all of the tests pass, someone will need to review the code and then merge it into the master branch. Once added into the master branch, the code will be automatically deployed.

Where can I view the website?

Once the code has been successfully built and deployed, under the “Actions” tab there will be a tab labeled as “Deploy to Firebase.” Clicking on that tab will display a dropdown description. Within the description there will be a bolded text “Hosting URL,” which is followed by a link. Copy and pasting the link will display the website.

Where do I find the result of the test?

Make sure to hover over to the “Actions” tab and then proceed to search for the name of the commit. (Note: Clicking on the “Continuous Integration” tab will only display tests if it’s too hard to differentiate between testing vs. build & deploy workflows)

I found a feature that could make testing much easier, how can I suggest it to you guys?

Contact any of the leads or if it makes it easier, contact anyone on the build team and explain what the feature is that you would like to be added. We will gladly look into adding more features into the pipeline to make things easier.

My test failed and there is little information on why it’s failing. How can I find more details?

Click on the name of the commit that failed. You will see a bunch of check marks and x marks. Click on the label that has a x marks to get a more detailed description as to why it failed.

Where can I learn more about the technologies we are using?

Firebase (Backend): https://firebase.google.com/
Webpack (Building Framework): https://webpack.js.org/
Github Action (CI/CD pipeline): https://github.com/features/actions
Mocha (Testing Framework): https://mochajs.org/